
Object Colors

Release 1.0.1

Oct 05, 2019

Contents

1	object_colors module	3
	Python Module Index	7
	Index	9

Adding colours to Python simplified into a single class

`pip install object-colors`

The source code is available [here](#)

CHAPTER 1

object_colors module

object_colors

This is a simple to use module designed to print to the terminal in color with minimal setup and instantiation

```
class object_colors.Color(*args, **kwargs)
```

Bases: object

Instantiate object with all attributes or set later

Set class attributes for Color.print() or Color.get()

effects: 'none', "bold", 'bright', "underline", "negative"

colors: "black", "red", "green", "yellow", "blue", "purple", "cyan", "white"

Parameters

- **args** – Single digit ANSI Escape code (second) or string arguments Codes can be entered in any number of arguments - ordered - and the first three integers/strings will be used args take precedence over kwargs
- **kwargs** – Three specific keys can be used for class attributes: text, effect or background Subclasses of any name can be defined

```
keys = ['text', 'effect', 'background']
```

```
pop (select: str) → Union[str, dict, None]
```

Remove keypair from __dict__ and return to variable

Parameters **select** – Key to remove

Returns Class dict or None

```
print (*args, **kwargs) → None
```

Enhanced print function for class and subclasses

Parameters

- **args** – Variable number of strings can be entered if syntax allows Method behaves just like builtin print()
- **kwargs** – builtin print() kwargs

get (*string: str*) → str

String can be returned as variable(s) for assorted color printing or can be printed directly by calling self.print()

Parameters **string** – String to print

Returns Colored string

static opts (*key: str*) → list

Get allowed values paired with text, effect and background

Parameters **key** – Specific key for allowed values

Returns effects for effect and colors for text and background

static assign_kw (*key: str, kwargs: dict, opts: list, default: int*) → dict

First statement returns kwargs as is if valid escape code provided The second will assign the default value to kwargs if invalid value is provided Otherwise, if keypair is string - but valid - value will be converted to integer

Parameters

- **key** – Allowed keys: text, effect, background
- **kwargs** – Original user defined kwargs or empty dict
- **opts** – Valid options corresponding with key
- **default** – Default integer value if arg invalid or not entered

Returns kwargs with default values, original args or string converted to an integer

get_processed (*args: Any, kwargs: dict*) → dict

Organise args and kwargs into a parsable dictionary

Parameters

- **args** – User defined arguments: list(s) of integers or tuple of strings
- **kwargs** – Keyword arguments for class attributes or subclass attributes

Returns Dictionary for class attributes

make_subclass (*args: Any, kwargs: dict*) → bool

Compile the dictionary to be used as values for new class Returned list and dict will replace the arg and kwarg parameters

Parameters

- **args** – Class args for Color subclass
- **kwargs** – Class kwargs for Color subclass

Returns Validated dictionary for class __dict__

class_kwargs (*args: Any, kwargs: dict*) → None

static process_args (*args: Any*) → list

e.g. instead of text="red", effect="bold", background="blue" 114 would get the same result

Parameters **args** – Integer list or individual args

Returns List to later be iterated through for arguments

static get_nest_dict (*key: str, args: list, kwargs: dict*) → dict

Run through corresponding allowed keys against the length of the processed args list and assign ordered list indices to each key Kwargs will be Color's kwargs argument for class instance subclass

Parameters

- **key** – Allowed keys: text, effect, background
- **args** – Validated user arguments
- **kwargs** – Kwargs dict[key] to add nested subclass dict to

Returns Kwargs for subclass __dict__

class_ints (*kwargs: dict*) → dict

Resolves values which may not be entered as tuples, and therefore will confuse the class methods which are expecting args

Parameters **kwargs** – Subclass dictionary

Returns Subclass dictionary with processed tuple as args

set (**args, **kwargs*) → None

Call to change/update/add class values or add subclasses for new text, effects and backgrounds

Parameters

- **args** – Integer escape codes or strings to be converted into escape codes
- **kwargs** – More precise keyword arguments

- genindex

This documentation was last updated on Oct 05, 2019

O

`object_colors`, 3

A

`assign_kw()` (*object_colors.Color static method*), 4

C

`class_ints()` (*object_colors.Color method*), 5

`class_kwargs()` (*object_colors.Color method*), 4

`Color` (*class in object_colors*), 3

G

`get()` (*object_colors.Color method*), 4

`get_nest_dict()` (*object_colors.Color static method*), 4

`get_processed()` (*object_colors.Color method*), 4

K

`keys` (*object_colors.Color attribute*), 3

M

`make_subclass()` (*object_colors.Color method*), 4

O

`object_colors` (*module*), 3

`opts()` (*object_colors.Color static method*), 4

P

`pop()` (*object_colors.Color method*), 3

`print()` (*object_colors.Color method*), 3

`process_args()` (*object_colors.Color static method*), 4

S

`set()` (*object_colors.Color method*), 5